Szczepan Polak
Mateusz Wojcik
Majka Miezianko

LARGE DATASETS EMBEDDING AND VISUALIZATION

Abstract *Visualization plays a crucial role in understanding complex and high-dimensional data by revealing patterns and relationships that may not be readily apparent. It serves as both a fundamental aspect of data analysis and an independent field within machine learning. By visualizing data, we can identify clusters and similarities among observations, gaining valuable insights and intuition. In the case of multidimensional and high-dimensional data, it becomes necessary to reduce the dimensions to a manageable level, typically three. However, linear methods like PCA may not capture the nonlinear relationships effectively. Therefore, Manifold Learning techniques are employed to uncover the underlying surface or manifold on which the data points lie, allowing for meaningful projections into lower-dimensional spaces of choice.*

Keywords IVHD, PaCMAP, UMAP, TriMAP, t-SNE

1. Introduction, Motivation

Within the framework of this study, the goal was to examine and compare among themselves the most popular methods of visualizing multidimensional data:

- t-SNE
- Umap
- TriMAP
- PacMAP
- IVHD

As part of the research, three simple datasets based on geometric figures were generated, with different variants of density and dimensionality.

The main goal of the project was to check how the above algorithms behave depending on the parameters of the sets and, in particular, whether they separate well different classes of points, including how this changes with increasing dimensions, and whether they correctly reflect the real patterns of data in a two-dimensional space.

2. Datasets description

For the purposes of the study, 3 types of synthetic data sets were prepared. Datasets consist of combinations of figures:

- Torus
- Sphere
- Ball

Each type of dataset, has the ability to create its counterpart in N dimensional space. Table 1 shows the dataset type along with its visualization in 2 and 3-dimensional space.

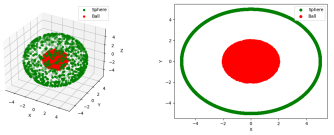
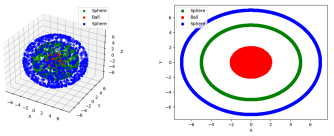
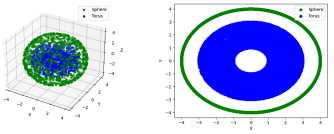
Types of datasets descriptions:

- Dataset type 1 consists of a ball with a radius (R) equal to 2 and a sphere with a radius (R) equal to 5.
- Dataset type 2 consists of a ball with radius (R) equal to 2 and a sphere with radius (R) equal to 5 and a second sphere with radius (R) equal to 7.
- Dataset type 3 consists of a torus with radius (R) equal to 2 and radius (r) equal to 1, and a sphere with radius (R) equal to 4.

For the purpose of the tests, various combinations of the dataset types described earlier were created with different amounts of number of points and dimensionality, according to the:

- number of dimensions [3, 5, 10, 20],
- number of points [1000, 2000, 5000, 20000, 50000].

Table 1
Table with dataset names and figures

Dataset Name	3D and 2D example
Ball inside sphere	
Ball inside two spheres	
Torus inside sphere	

3. Dimensionality reduction methods

3.1. IVHD

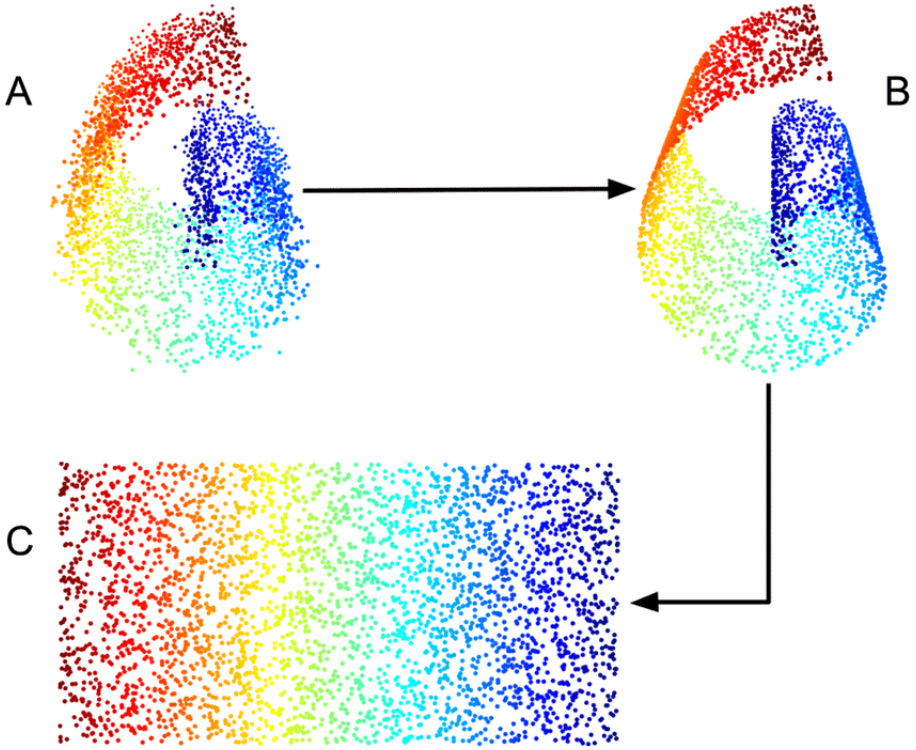
Interactive Visualization of High-Dimensional Data (IVHD) is a state-of-art and efficient method for visualizing high-dimensional data. It is based on a simplified force-directed implementation of Multidimensional Scaling (MDS). IVHD aims to achieve linear-time and memory complexity by using a limited number of distances from the high-dimensional space and their corresponding distances in the low-dimensional space.

The first step in IVHD involves creating a graph of nearest neighbors, known as the nearest neighbor graph (nn-graph). The key assumption underlying IVHD is that a small number of nearest neighbors (nn) is sufficient to approximate the underlying data manifold effectively. However, using a small number of nearest neighbors often leads to a sparsely connected nn-graph. To address this, IVHD also connects randomly selected neighbors (rn) to ensure better connectivity.

The goal of IVHD is to obtain a 2D embedding of the data by minimizing a stress function. This stress function is derived from an equation that captures the discrepancy between the pairwise distances in the high-dimensional space and their corresponding distances in the low-dimensional embedding. This stress function is similar to the one used in classical Multidimensional Scaling (MDS).

In order to reduce memory usage and computational complexity, IVHD employs binary distances between data vectors instead of full distance calculations. This allows for more efficient processing of the high-dimensional data.

While the results produced by IVHD are generally good, it should be noted that the local neighborhood structure obtained by this method may not be as accurate as that achieved by other Similarity Network Embedding (SNE) variants. Nevertheless, IVHD is still a promising approach due to its favorable computational and memory complexities, as well as its requirement of only a small number of parameters.[3].



3.2. PaCMAP

PaCMAP (Parallel Constrained MAP) is a dimensionality reduction technique that aims to preserve the global structure of high-dimensional data in a low-dimensional space. It is a nonlinear method that combines concepts from t-SNE (t-Distributed Stochastic Neighbor Embedding) and UMAP (Uniform Manifold Approximation and Projection).

The PaCMAP algorithm operates in two main steps: the construction of the graph and the optimization of the embedding. In the graph construction step, PaCMAP builds a k-nearest neighbors graph based on the input data. The number of neighbors, denoted by k, is a parameter that determines the local neighborhood size for each data point. This graph represents the local relationships between the data points.

In the optimization step, PaCMAP seeks to find a low-dimensional representation of the data that preserves the global structure. It minimizes a cost function called

the Constrained MAP objective, which combines a measure of the preservation of local neighborhoods and the preservation of global distances. By balancing these two objectives, PaCMAP aims to produce an embedding that captures both local and global patterns.

PaCMAP introduces several innovations to improve the efficiency and scalability of the embedding process. It employs parallel computing techniques to accelerate the computations, making it suitable for large-scale datasets. It also incorporates constraints to control the embedding process, such as enforcing a minimum distance between points [4].

One notable advantage of PaCMAP is its ability to handle high-dimensional data efficiently. It can capture nonlinear relationships and reveal complex structures in the data. However, like other dimensionality reduction techniques, the interpretation of the resulting low-dimensional representation may be challenging, as the embedded dimensions do not have direct semantic meaning.

3.3. UMAP

The UMAP algorithm begins by constructing a fuzzy topological representation of the data. It creates a fuzzy simplicial complex, which is a graph with edge weights indicating the likelihood of two points being connected. The construction of the initial high-dimensional graph involves determining a radius for each point, and connecting points whose radius overlap. The radius is chosen locally based on the distance to each point's nearest neighbors, ensuring that each point is connected to at least one closest neighbor. This fuzzy topological representation is then used to convert the data into a low-dimensional representation by optimizing the layout to be as similar as possible.

UMAP provides control over the balance between local and global structure through its parameters. The most influential parameter is n_neighbors, which determines the size of the local neighborhood considered by UMAP in learning the manifold structure. Lower values of n_neighbors focus more on local structures, while larger values incorporate more global structures. The second important parameter is min_dist, which controls the compactness of points in the low-dimensional space. It specifies the minimum distance allowed between points in the representation.

Although UMAP yields impressive results, it also has some drawbacks. One limitation is its lack of interpretability, as the dimensions in the embedding space do not have specific meanings like in Principal Component Analysis (PCA), for example. UMAP can also find manifold structures within the noise of a dataset. When working with a small sample of noisy data, the structures obtained from the noise are more prominent. However, as more data is sampled, the influence of noise tends to decrease, revealing more meaningful structures [2].

3.4. TriMAP

TriMap is a dimensionality reduction method that uses triplet constraints to form a low-dimensional embedding of a set of points. The triplet constraints are of the form "point i is closer to point j than point k". The triplets are sampled from the high-dimensional representation of the points and a weighting scheme is used to reflect the importance of each triplet.

TriMap provides a significantly better global view of the data than the other dimensionality reduction methods such t-SNE, LargeVis, and UMAP. The global structure includes relative distances of the clusters, multiple scales in the data, and the existence of possible outliers [1].

3.5. t-SNE

t-SNE t-distributed stochastic neighbor embedding - is a nonlinear dimensionality reduction technique used to visualize multidimensional data, usually in 2 or 3-dimensional space. The algorithm is mainly based on the method of ordering neighbors using t-distribution. The operation of the algorithm can be divided into 2 main steps. The first is to calculate the probability distribution for pairs of points in a high-dimensional space, so that similar points are assigned a higher probability and not similar points a lower probability. The next step is to perform the same operation only in the target low-dimensional space. The final step is to minimize the Pullback Leibler divergence from the calculated probabilities.

4. Metrics

To evaluate dimensionality reduction (DR) techniques and assess their ability to preserve both local neighborhood information and global structure, metrics and measures are essential. Multiple metrics provide a comprehensive understanding of the quality of data separation. During the evaluation process of a dimensionality reduction technique, it is crucial not only to rely on visual interpretation, but also to examine the metric values in order to compare various embedding methods.

4.1. DR quality and KNN gain

DR quality is a metric that measures the fidelity of the neighborhood representation in the reduced-dimensional space. It is derived from the co-ranking matrix, which compares the pairwise distances between data points in the original high-dimensional space and their counterparts in the reduced-dimensional space. The DR quality metric provides an assessment of how well the local structure and relationships among neighboring points are preserved after dimensionality reduction. A higher DR quality value indicates a better preservation of the neighborhood information.

KNN gain, on the other hand, quantifies the improvement in preserving the class information within the neighborhoods. It measures the fraction of additional samples from the same class that are contained within a given neighborhood size

in the reduced-dimensional space compared to the original high-dimensional space. KNN gain provides insights into the ability of a dimensionality reduction technique to maintain the local structure in terms of class separability. A higher KNN gain value signifies a better preservation of class information within the neighborhoods.

Both DR quality and KNN gain are useful in evaluating the performance of dimensionality reduction techniques, as they provide quantitative measures of how well the techniques preserve the local neighborhood structure and class information in the reduced-dimensional space.

4.2. Shepard diagram

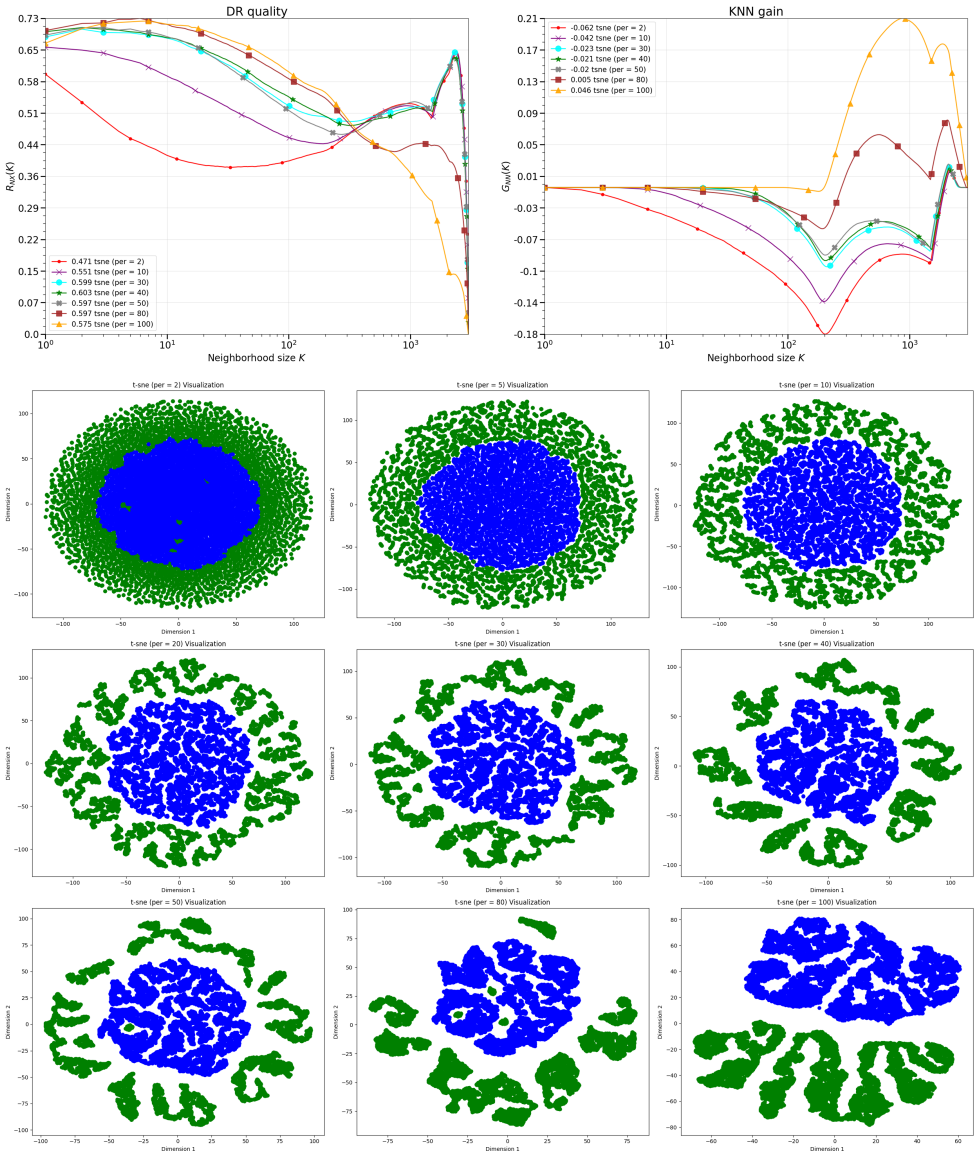
The Shepard diagram is a graphical representation, in the form of a scatter plot, that illustrates the distances between points in the original high-dimensional space and the reduced-dimensional space. Typically, the x-axis of the Shepard diagram represents the distances between points in the original space, while the y-axis represents the distances after dimensionality reduction. In an ideal scenario, where the dimensionality reduction technique accurately preserves the distances, the points on the scatter plot would form a straight line.

However, since dimensionality reduction involves the lossy compression of information, it is rare to observe a perfectly straight line in the Shepard diagram. The presence of deviations and dispersion around the straight line indicates a weaker preservation of distances between the samples in the original space and the reduced space.

5. Visualizations and comparison between dimensionality reduction techniques

5.1. Parameters selection

Prior to the use of dimensionality reduction techniques, tests were performed to select the optimal hyperparameters for each of the dimensionality reduction techniques tested. Not all of the results obtained were inconclusive, and in some cases the opposite of those expected from the data visualization technics. In addition, the authors found that it would be necessary to select parameters for each combination of dataset, reduction method, number of dimensions and number of points separately, which would entail redesigning multiple tests. Which, due to limited resources, time and the need for high computing power, was not done. The final tests were performed within the standard settings of each method. Figures down below include an example of the results of t-SNE different perplexities, a sphere in a sphere.



5.2. Ball inside sphere

The visualization of embeddings of a ball inside a single sphere should allow for a comprehensive understanding of the geometric arrangement of the data points. By reducing the dimensionality of the data and representing it visually, patterns, clusters, and the overall spatial relationships within the sphere can be effectively analyzed and interpreted.

5.2.1. t-SNE

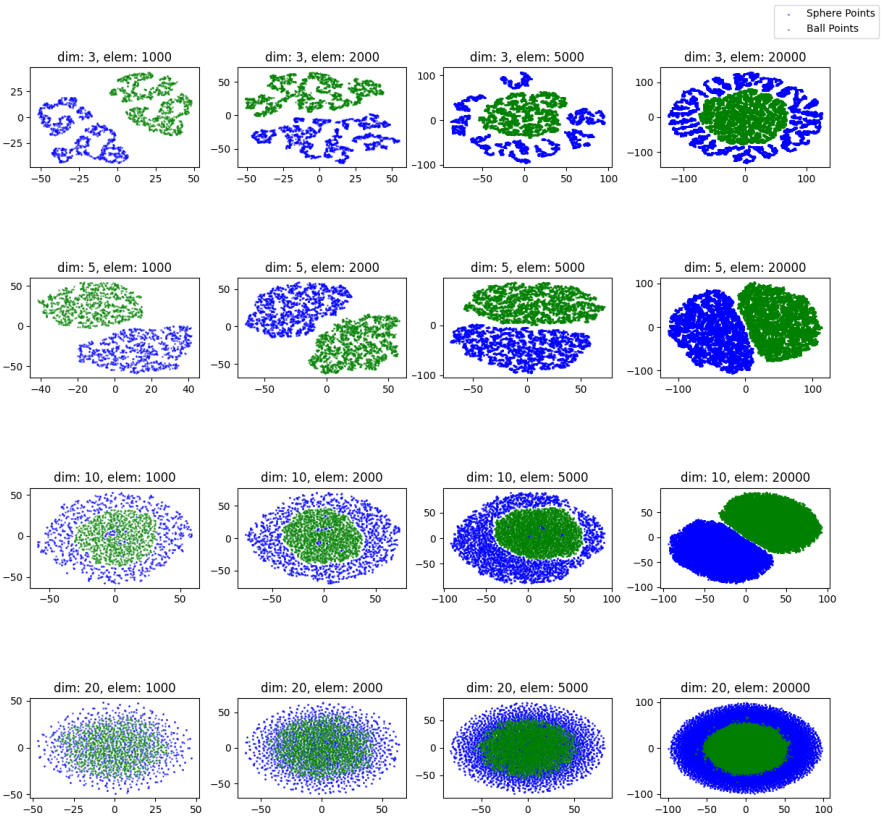


Figure 1. Ball inside sphere - results of t-SNE

The t-SNE handles the separation of classes into ball and sphere reasonably well. At low densities, as dimensionality increases, there is a mixing of classes however, densifying the points significantly improves the results. This algorithm in some cases was also able to rearrange the global structures, thus the expected conformal shape, the real mapping in 2 dimensions.

5.2.2. UMAP

Umap also correctly separates the two classes of points, although it does not deal with the actual representation of the structure. It should be noted that visualizations in the case of 20 dimensions do not look the best, and the classes blend together, moreover, the density of points does not completely solve this problem.

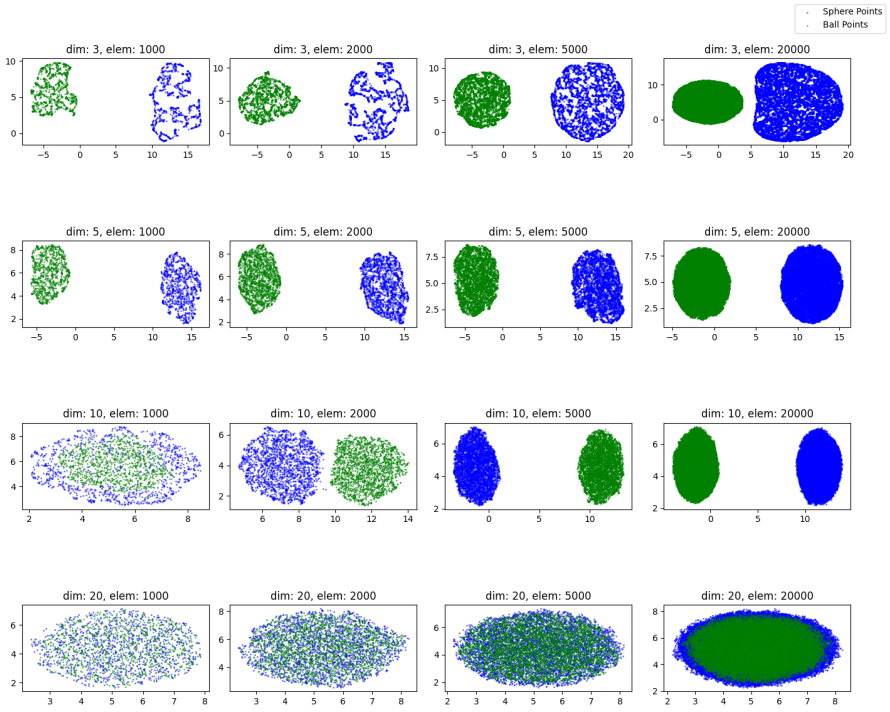


Figure 2. Ball inside sphere - results of Umap

5.2.3. PaCMAP

PaCMAP, although it definitely obtained the expected result in the case of visualizing the reduction from 3 dimensions for 20000 points, has some problems in other cases. Although it can correctly separate 2 classes from each other, especially in lower dimensionality, it had a problem with the sphere, which in many cases was split into two parts.

5.2.4. Tri-MAP

Trimap correctly separates the two classes of objects. Particularly good, and as expected, are the visualizations of the reduction from three dimensions. As the dimensionality increases, although it gives reasonably correct results in the cases studied, it is noticeable that the algorithm begins to have some problems.

5.2.5. IVHD

IVHD does not always separate correctly the two classes of points and it does not deal with the actual representation of the structure. Interestingly, it should be noted

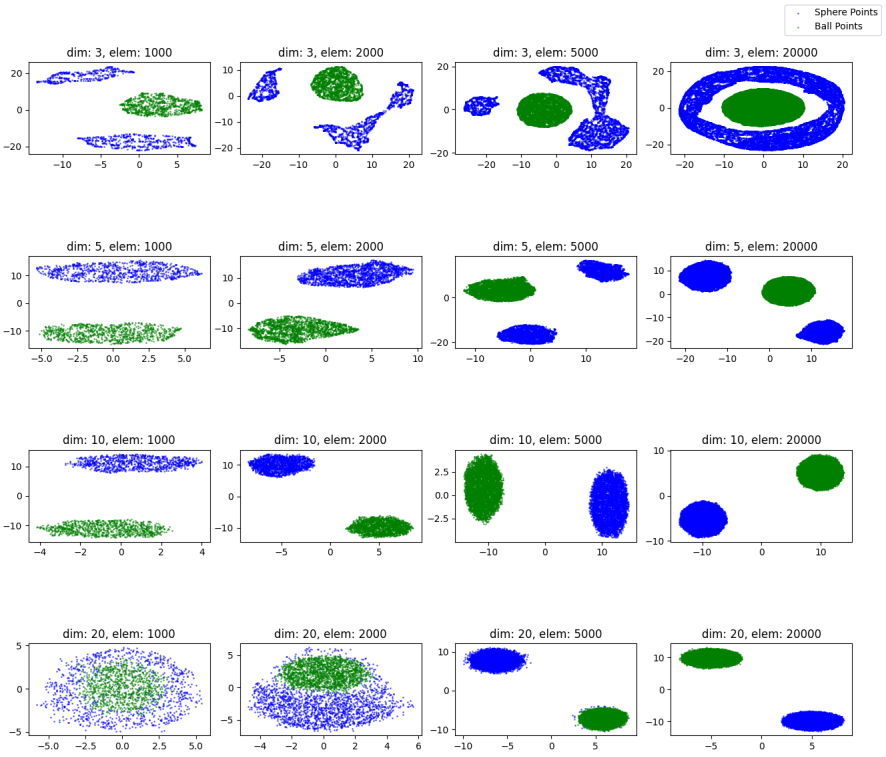


Figure 3. Ball inside sphere - results of PaCMAP

that visualization in the case of 30 dimensions seems the most accurate in comparison to lower dimensional space.

5.2.6. DR quality and KNN gain

Analyzing the metrics, one can see that in most cases looking at the local structure, all methods achieve similar results. However, t-SNE definitely stands out, achieving noticeable differences. As a rule, the noticeable differences between the algorithms become apparent in the global structure where they reach their maxima. The two metrics are somewhat mutually exclusive in the comparison between them, so that the best methods in terms of Dr quality are not necessarily the best in terms of KNN gain as well. One can get the impression that, on average, it is t-SNE that fits best in the various situations tested, as an algorithm that may not perform best, but holds a certain average. Looking only at Dr. quality, PacMAP and TriMAP perform best, while analyzing only KNN gain the results are similar, with the observation that in the case of reduction from 20 dimensions t-SNE performs much better than the others.

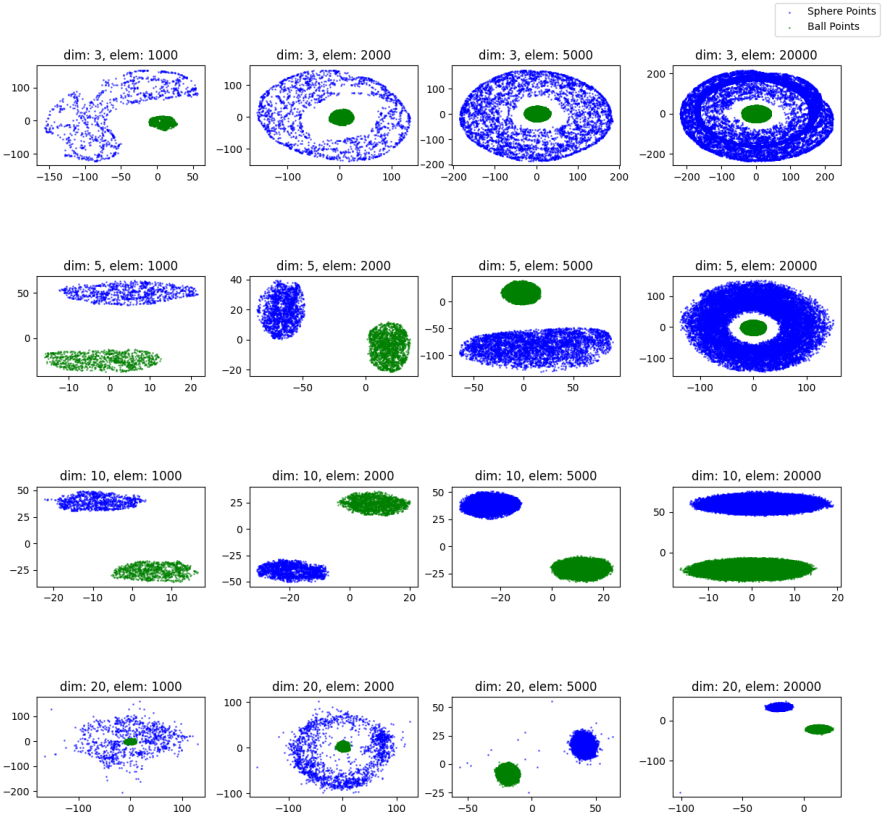


Figure 4. Ball inside sphere - results of Trimap

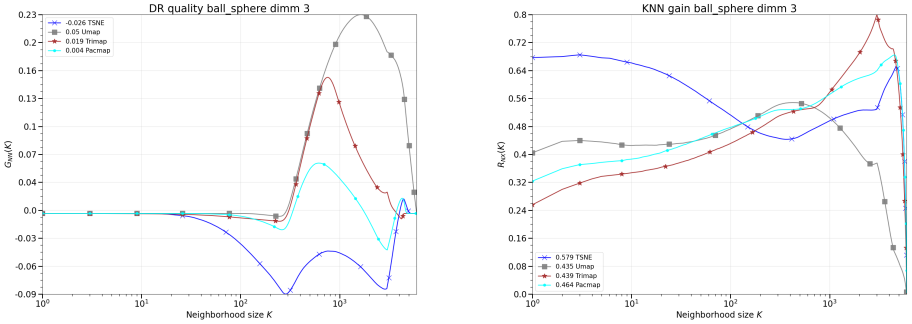


Figure 5. Ball inside sphere - DR quality and KNN gain - 3 dim

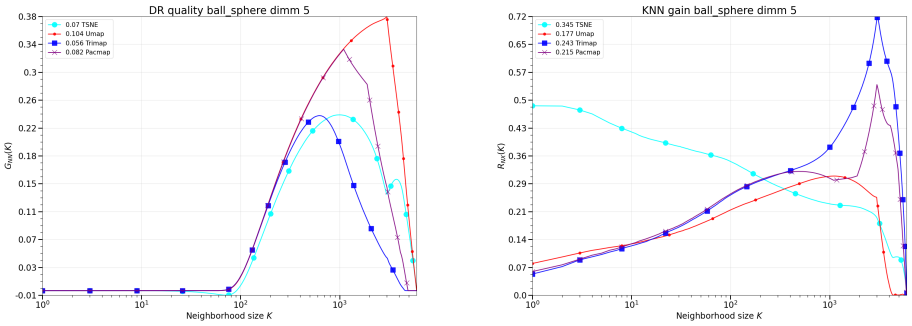


Figure 6. Ball inside sphere - DR quality and KNN gain - 5 dim

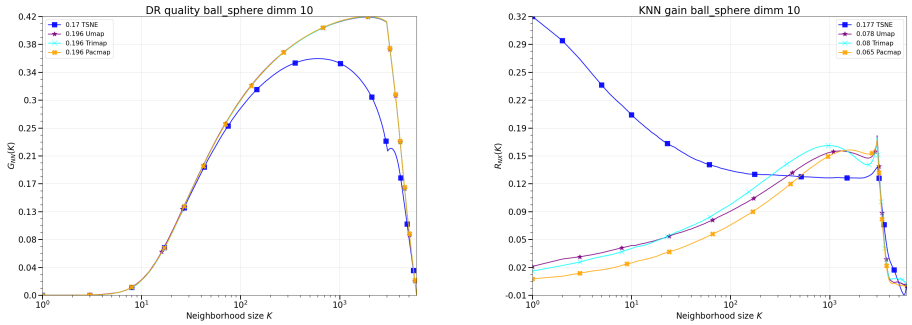


Figure 7. Ball inside sphere - DR quality and KNN gain - 10 dim

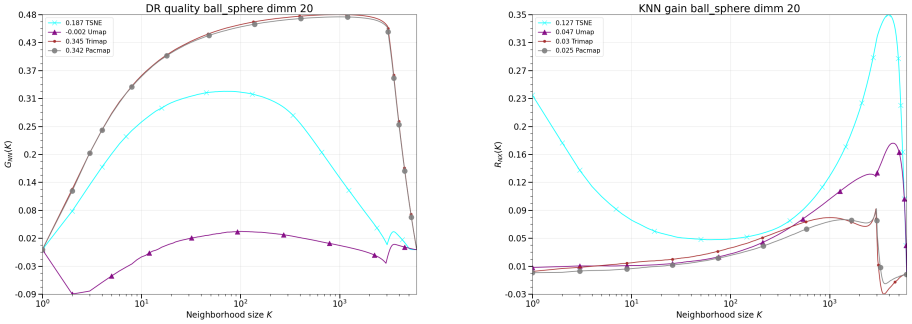


Figure 8. Ball inside sphere - DR quality and KNN gain - 20 dim

5.3. Ball inside two spheres

The visualization of embeddings of a ball inside two spheres should provide a clear representation of the spatial arrangement and relationships between the data points. By mapping the high-dimensional data onto a lower-dimensional space, the visualizations enable the identification of patterns, clusters, and the overall structure of the data within the context of the ball and spheres.

5.3.1. t-SNE

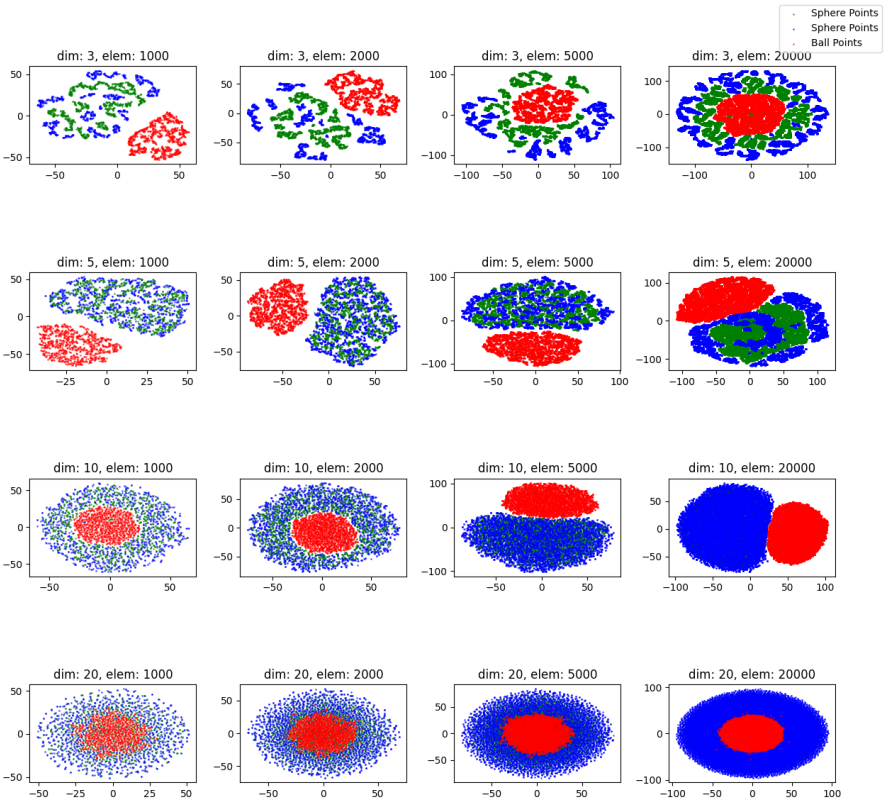


Figure 9. Ball two sphere - results of t-SNE

t-SNE, failed to correctly separate classes of points from each other. Although in most cases the separation of the sphere was correct, problems were posed by spheres that blended together. The only close approximation to the expected visualization is the visualization of the reduction from three dimensions in the case of 20000 points.

5.3.2. UMAP

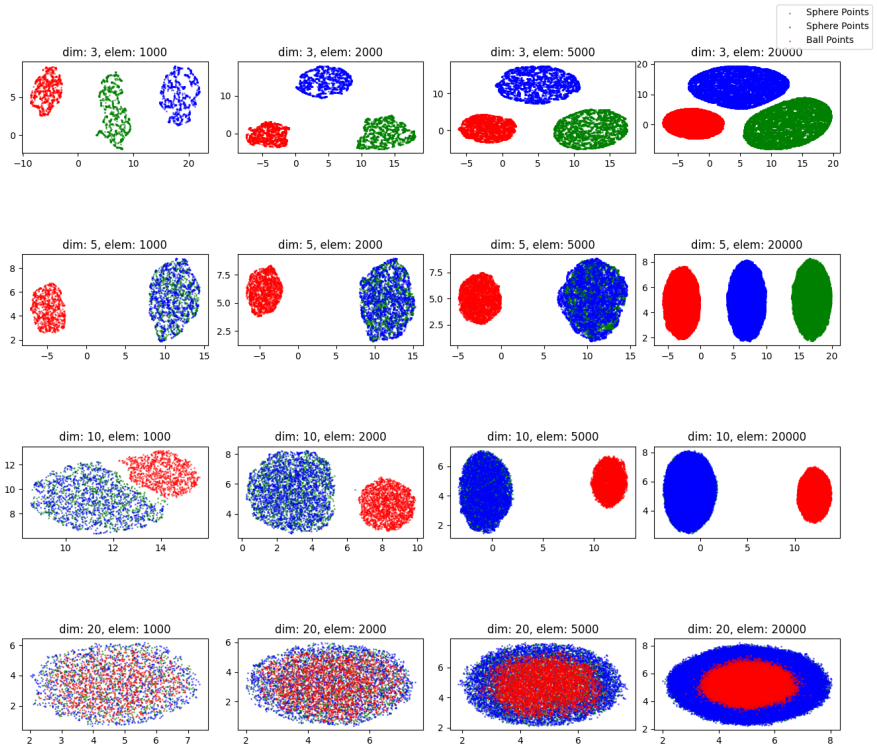


Figure 10. Ball two sphere - results of Umap

Umap, in the case of low dimensionality and adequate density, was able to separate all classes of points, but it should be noted that they did not represent the desired global structure. The increase in dimensionality caused the classes, particularly spheres, to blend together. Thickening the points did not add much to the improvement.

5.3.3. PaCMAP

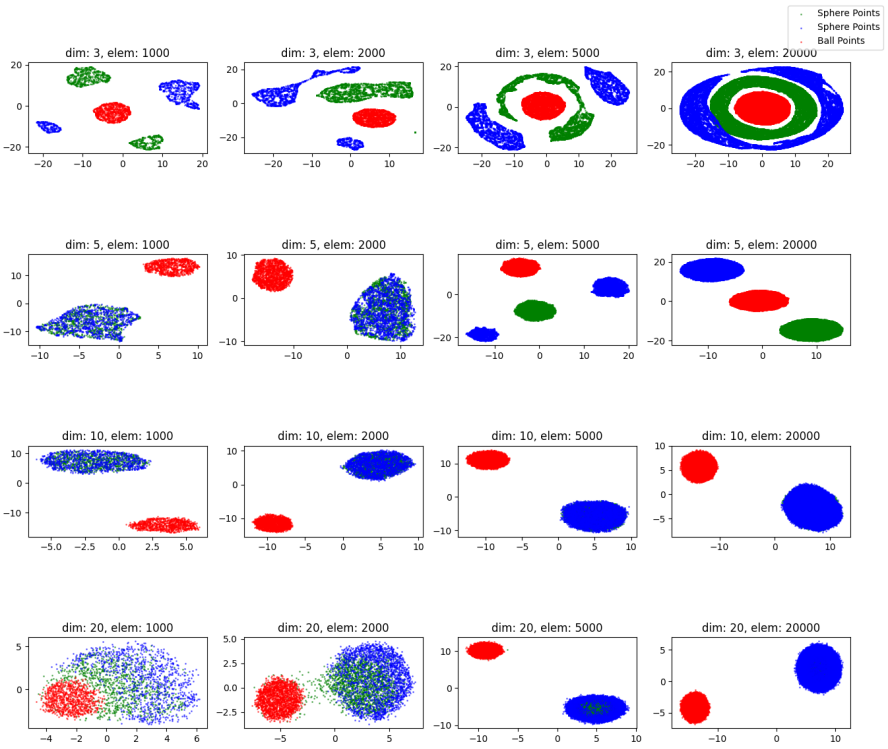


Figure 11. Ball two sphere - results of PaCMAP

PacMAP again achieved very good visualization in the case of reduction from three dimensions for 20000 points. However, it had significant problems in other cases. At low dimensionality, it tended to split the spheres into two groups, while densification improved the results. At higher dimensionality, the merging of spheres into a single group of points is evident.

5.3.4. Tri-MAP

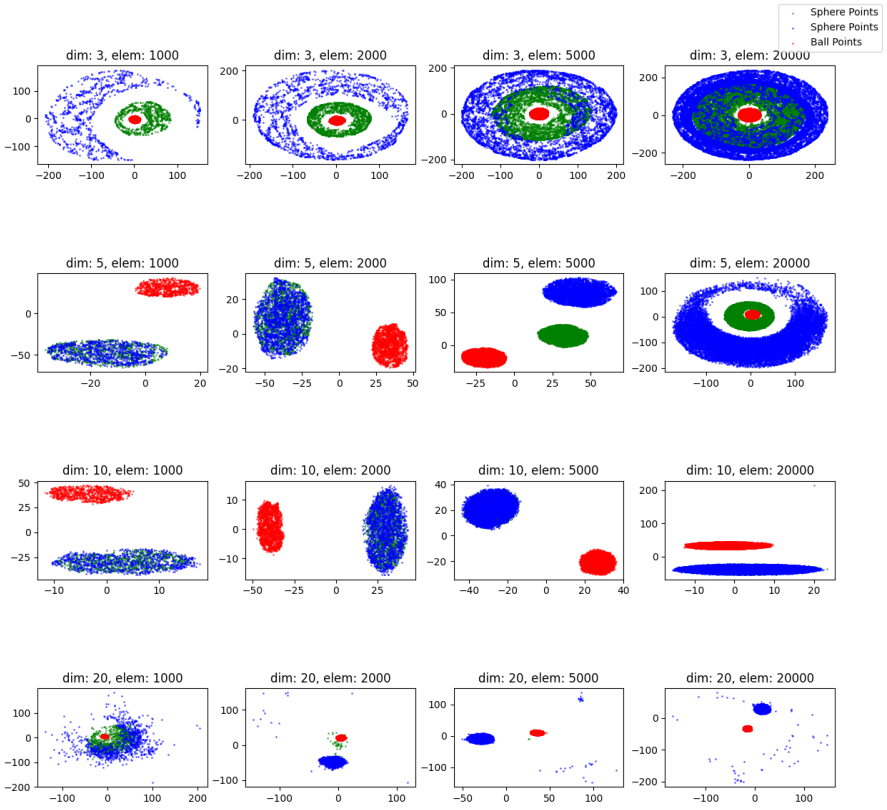


Figure 12. Ball two sphere - results of Tri-MAP

TriMAP also did not do the best. Interestingly, in the case of reduction from three dimensions, the results with medium density are better than those with higher density, where the points of both spheres begin to mix. The increase in dimensionality also in this case caused problems with the separation of the spheres from each other, and even noticeable announced problems of the algorithm.

5.3.5. IVHD

IVHD was able to separate all classes of points, but it should be noted that they did not represent the desired global structure. In every dimension, it caused the classes, particularly spheres, to blend together.

5.3.6. DR quality and KNN gain

Compared to the dataset with one sphere, we can notice some similarities as well as differences. First of all, it seems that the obtained results of the metrics show greater differences between the studied methods. As in the previous case, differences between KNN gain and Dr quality results are noticeable. Umap achieves the best results in terms of Dr quality, although the situation changes dramatically for 20 dimensions. In KNN gain, on the other hand, TriMAP and Pacmap generally achieve the best results, while for high dimensionality it is Umap and t-SNE that are significantly better.

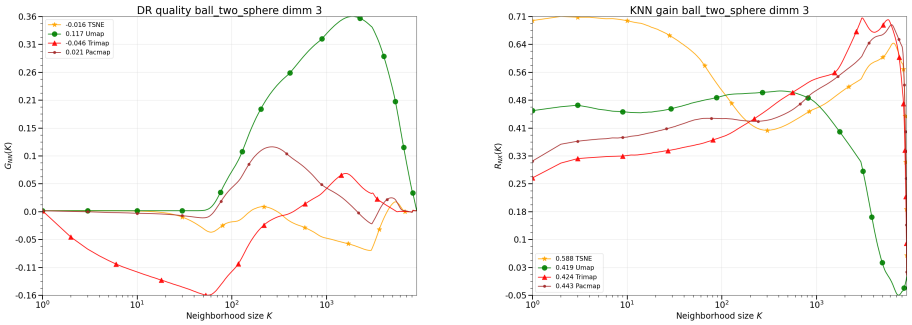


Figure 13. Ball two sphere - DR quality and KNN gain - 3 dim

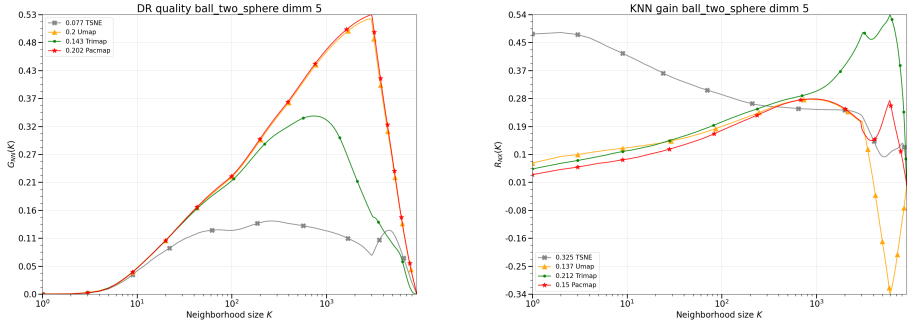


Figure 14. Ball two spheres - DR quality and KNN gain - 5 dim

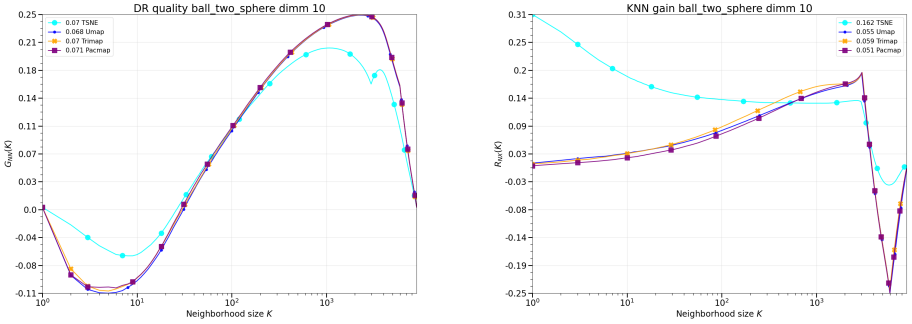


Figure 15. Ball two spheres - DR quality and KNN gain - 10 dim

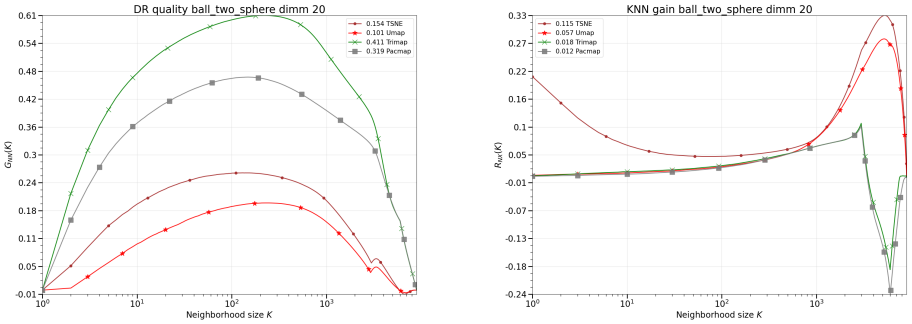


Figure 16. Ball two spheres - DR quality and KNN gain - 20 dim

5.4. Torus inside sphere

The visualization of embeddings of a torus inside a sphere provides a challenging task due to the complex interplay between the curved surfaces. This type of embedding requires careful representation and interpretation to capture the intricate spatial relationships and geometric properties within the torus-sphere configuration. It presents a more demanding visualization task compared to the embeddings of a ball inside a sphere or two spheres.

5.4.1. t-SNE

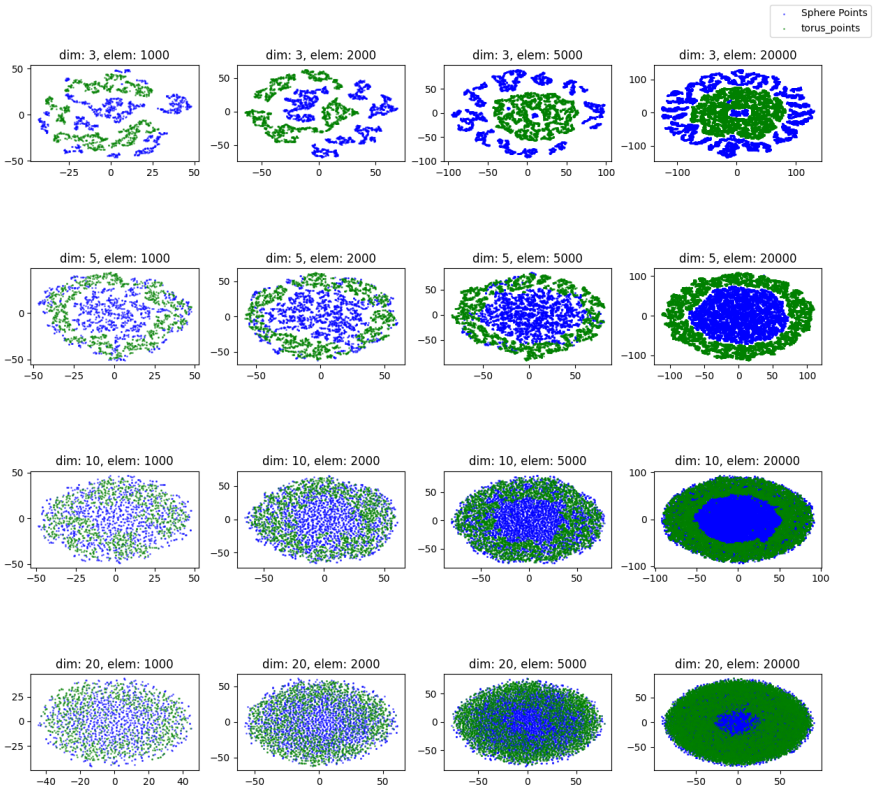


Figure 17. Torus inside sphere - results of t-SNE

t-SNE did not get good visualizations. The points mix, with each other and do not keep the expected shapes. E.g. in the case of reduction from 5 dimensions with 20000 points although, there is a noticeable separation of classes between each other this torus visualization indicates that it is the sphere that finds itself inside the torus while in, that reality the situation looks the opposite.

5.4.2. UMAP

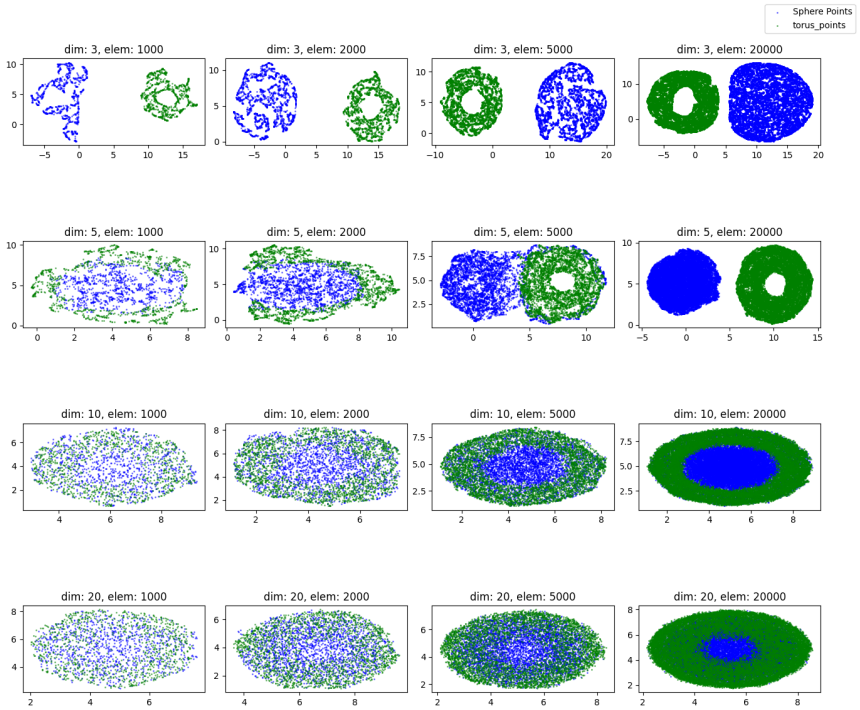


Figure 18. Torus inside sphere - results of UMAP

In the case of Umap with lower dimensionality and adequate density, visualizations look excellent. Both classes of objects are separated from each other, what's more, shapes such as torus shape are visible. Of course, this is not in accordance with the intended effect of conforming to the actual representation in 2 dimensions, but the very fact of the correct representation of the figure as a torus is definitely a plus. The increase in dimensionality causes the classes to begin to blend together, and the sphere as in the previous case is classified incongruously as if it were inside a torus.

5.4.3. PaCMAP

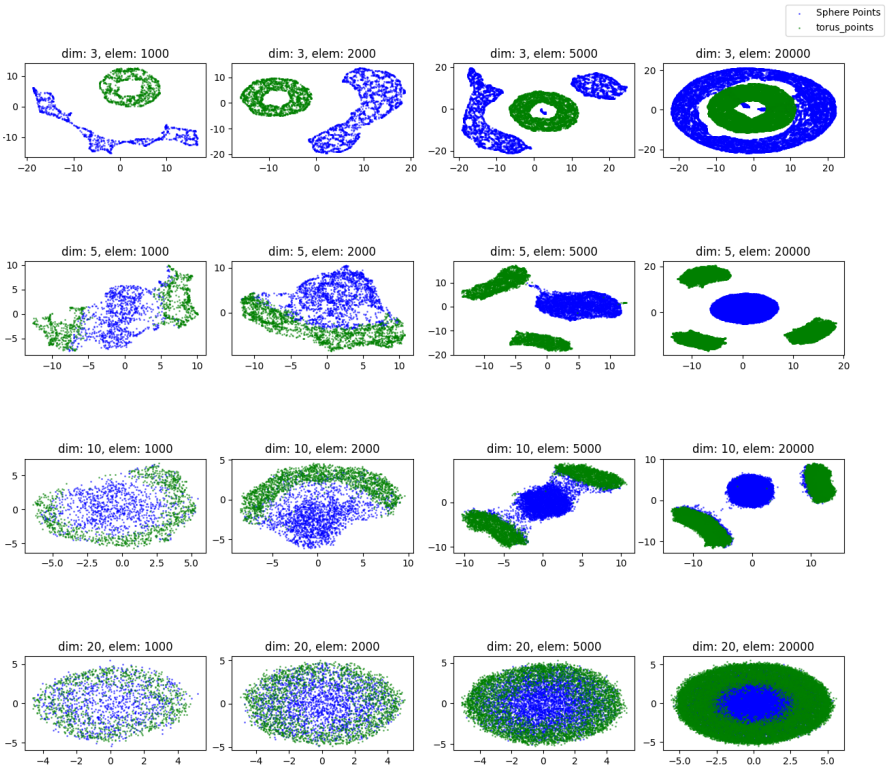


Figure 19. Torus inside sphere - results of PaCMAP

PacMAP, as in previous datasets, does not perform well. The main shortcoming again is the splitting of one class into parts, and the mixing of classes both at low density and at higher dimensionality. It seems that the only reasonably snatched up and close to intended is the visualization of the reduction from three dimensions with 20000 points.

5.4.4. Tri-MAP

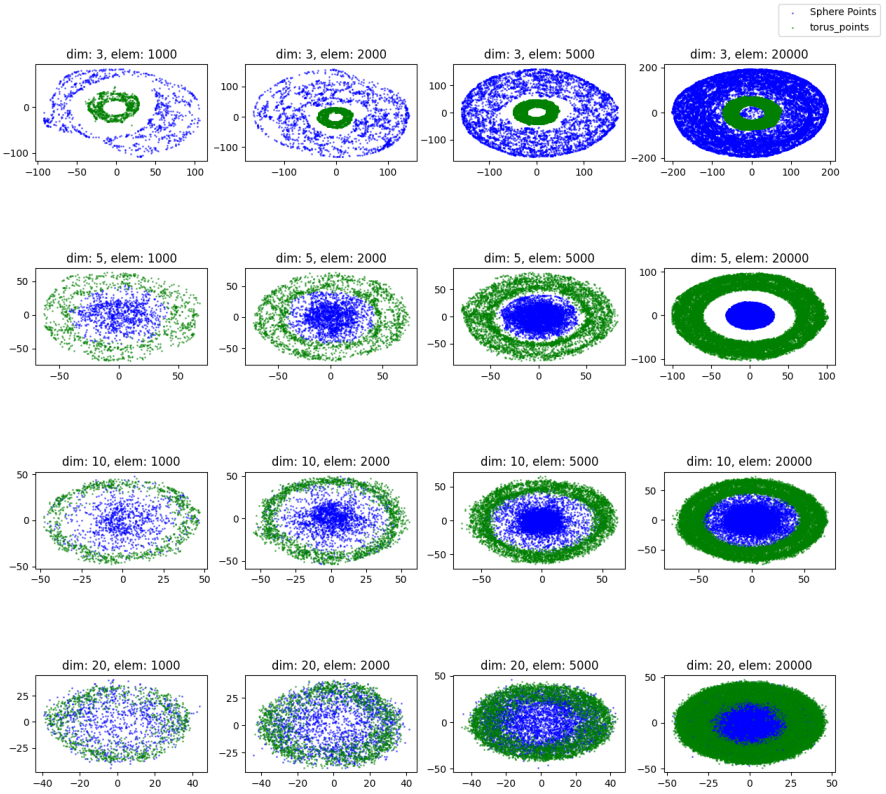


Figure 20. Torus inside sphere - results of TriMAP

TriMAP, when reduced from 3 dimensions, achieved reasonably good results, although it should be noted that increasing the density tended to affect the non-dimensionality. The increase in dimensionality resulted, as in other methods, in placing the sphere in the center of the torus. Nevertheless, it should be noted that, for example, in the reduction from 5 dimensions with 20000 points, the classes were clearly separated from each other. Obviously, increasing dimensionality caused more and more mixing of class points.

5.4.5. IVHD

IVHD, as in previous datasets, does not perform well. It seems that the only reasonably snatched up and close to intended is the visualization of the reduction from five dimensions with 5000 points.

5.4.6. DR quality and KNN gain

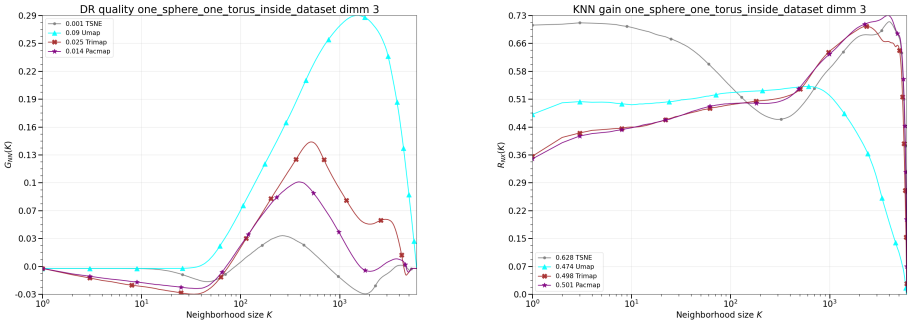


Figure 21. Torus inside sphere - DR quality and KNN gain - 3 dim

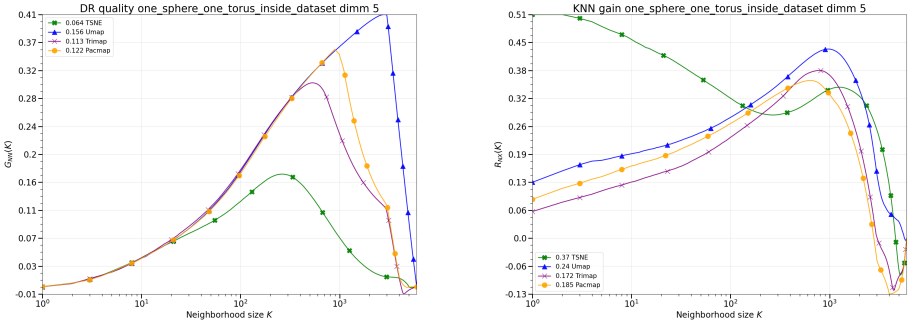


Figure 22. Torus inside sphere - DR quality and KNN gain - 5 dim

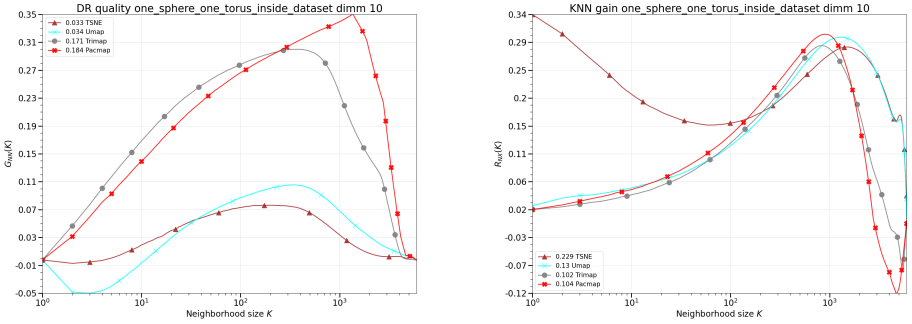


Figure 23. Torus inside sphere - DR quality and KNN gain - 10 dim

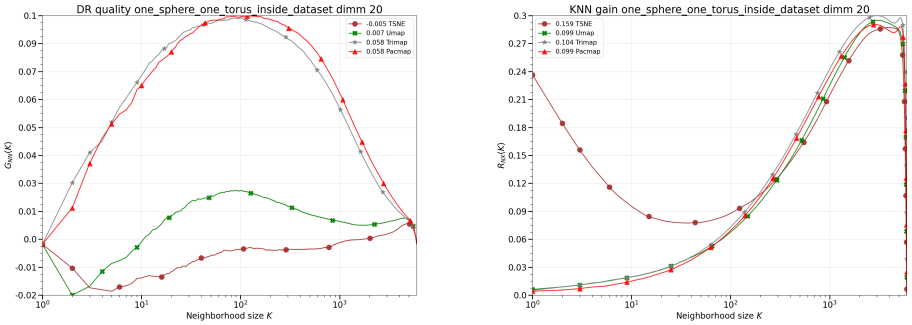


Figure 24. Torus inside sphere - DR quality and KNN gain - 20 dim

Looking at DR quality in the case of reduction from lower dimensionality, better results are achieved by Umap and t-SNE, on the other hand, as dimensionality increases, the situation reverses, and it is PacMAP and Trimap that achieve much better results. On the other hand, in the case of the KNN gain metric, the results of the tested methods are very close to each other, with the distinction that the outstanding method is t-SNE, which achieves significantly better results in a small neighborhood.

6. Conclusion

The results obtained turned out to be very different, in the counter of analyzed cases, datasets, dimensionality, density of points. Among the datasets analyzed, the dataset with two spheres and a sphere posed the greatest problems for the analyzed methods, where the biggest problem was the separation of the spheres from each other. The methods also had problems with a dataset with a torus, in particular with low density and high dimensionality.

Because of the diversity of results obtained, it is heavy to choose the best of the methods analyzed.

However, it seems that the best method that gave birth in many cases best in a small neighborhood is t-SNE, which in turn, as a rule, obtained the worst results in a large neighborhood.

Analysis of the metrics also indicates that PacMAP and TriMAP performed best in the large neighborhood in most cases. This is confirmed by the visualizations, particularly in low dimensionality, where the methods replicate the expected shapes of datasets in 2 dimensions quite well.

On the other hand, analyzing the visualizations in the context of Umap, we can see that in the case of high density and low dimensionality, it is very good at classifying groups.

It would be interesting to preprepare another study with higher density of the datasets, and see if this would improve the results in more dimensions.

It should be noted that the results of the analyzed metrics, are not consistent and direct comparison between them does not lead to unequivocal conclusions.

In case of analysed datasets, it is possible to visualize *crowding phenomenon* observed during dimension reduction by different embedding methods. Geometrical structure of figures assigned to 2-D space from high-dimensional space is usually distorted. Ball represents the highest class number (1 or 2) while spheres are assigned to lower class numbers (0 or 1). Then, we could see severe crowding phenomenon: ball class (1 or 2) is pushed dowards the center. This is due to the difference in norm concentration between high and low dimensions: a ball in higher dimensions has a volume that grows faster with radius. As a result, high-dimensional points are more likely to be distributed near the surface. When mapped to low dimensions, since there is less room near the surface, the points are pushed towards the center [3].

References

[1] Amid E., Warmuth M.K.: TriMap: Large-scale Dimensionality Reduction Using Triplets, 2022. URL http://dx.doi.org/10.48550/arXiv.1910.00204. ArXiv:1910.00204 [cs, stat].

[2] McInnes L., Healy J., Melville J.: UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction, 2020. URL http://arxiv.org/abs/1802.03426. ArXiv:1802.03426 [cs, stat].

[3] Minch B.: In search of the most efficient and memory-saving visualization of high dimensional data, 2023. URL http://arxiv.org/abs/2303.05455. ArXiv:2303.05455 [cs] version: 1.

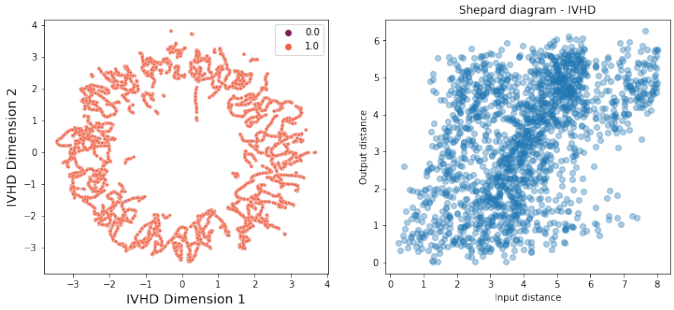
[4] Wang Y., Huang H., Rudin C., Shaposhnik Y.: Understanding How Dimension Reduction Tools Work: An Empirical Approach to Deciphering t-SNE, UMAP, TriMap, and PaCMAP for Data Visualization. In: .

Affiliations

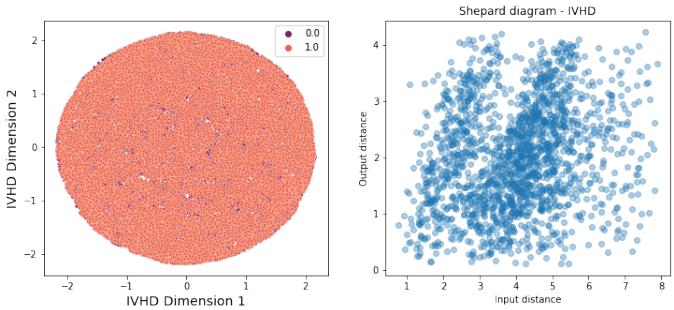
Szczepan Polak
AGH, address, szczepanpol@student.agh.edu.pl

Mateusz Wojcik
AGH, address, mateuszwojci@student.agh.edu.pl

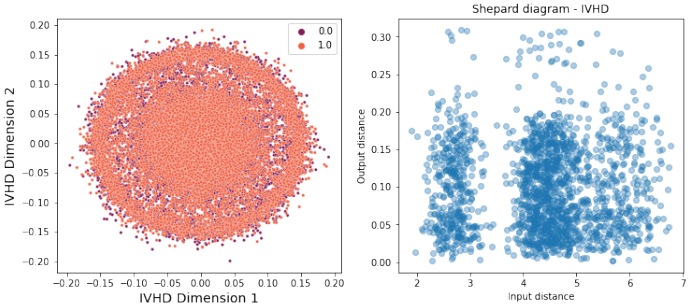
Majka Miezianko
AGH, mmiezianko@student.agh.edu.pl



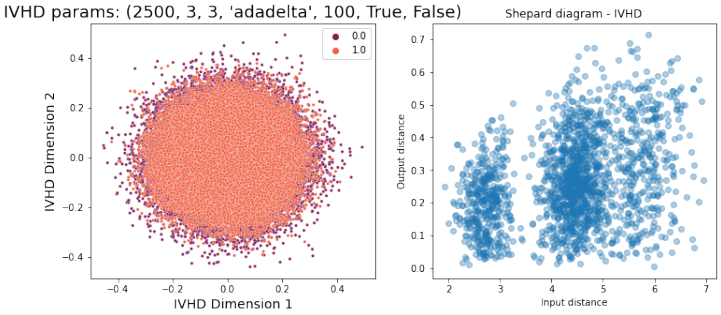
Ball inside sphere - results for IVHD for *nn_neighbours=7* i *random_neighbours=2* generated in 3-D space.



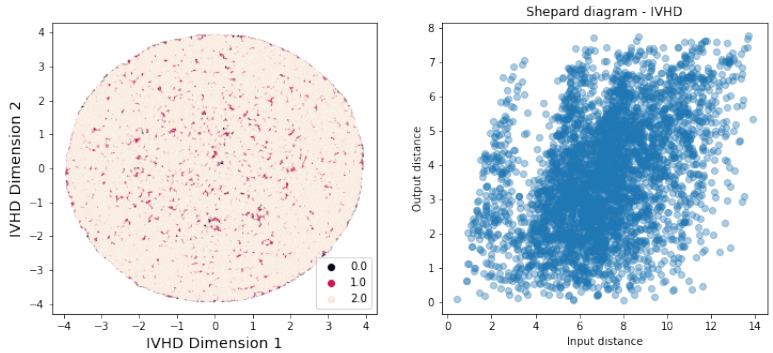
Ball inside sphere - results for IVHD for *nn_neighbours=7* i *random_neighbours=2* generated in 5-D space.



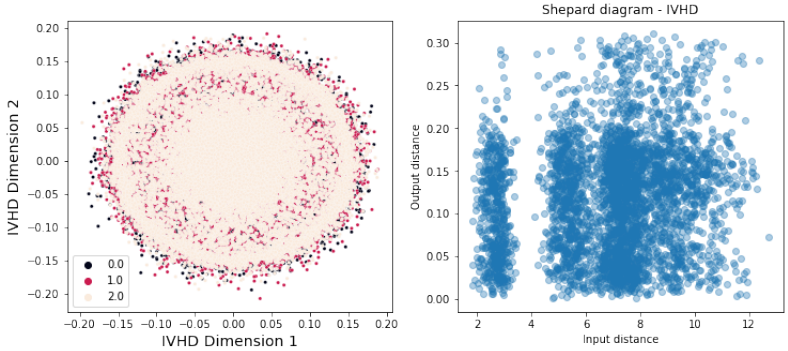
Ball inside sphere - results for IVHD for *nn_neighbours=7* i *random_neighbours=2* generated in 30-D space



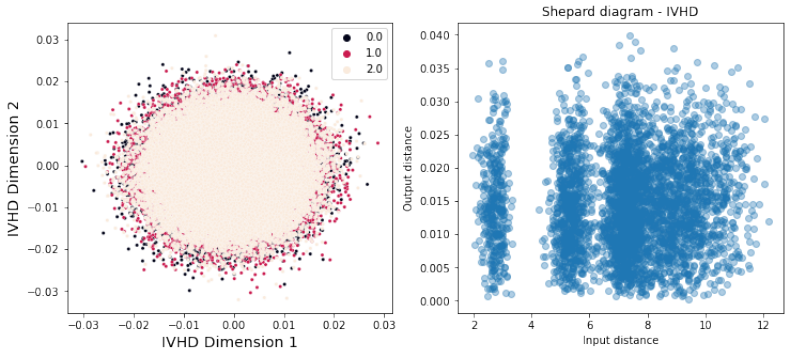
Ball inside sphere - results for IVHD for *nn_neighbours=3* i *random_neighbours=3* generated in 30-D space



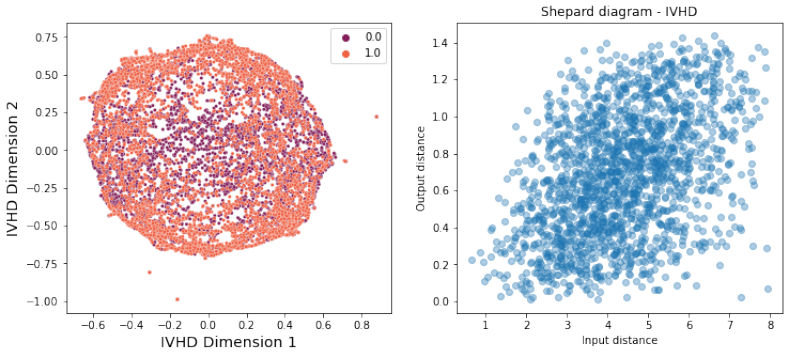
Ball inside two sphere - results for IVHD generated in 5-D space.



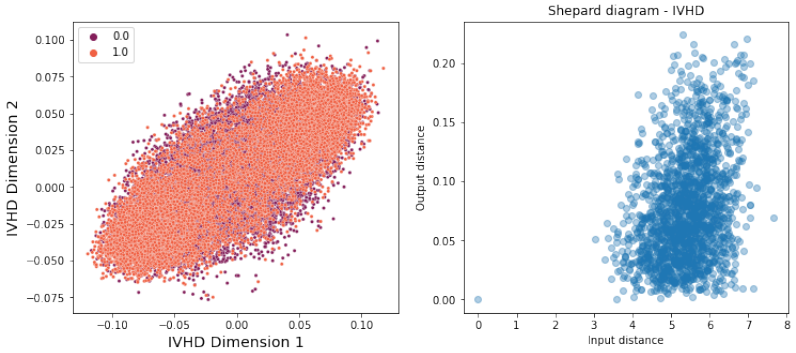
Ball inside two sphere - results for IVHD generated in 20-D space



Ball inside two sphere - results for IVHD generated in 30-D space



Torus inside sphere - results for IVHD generated in 5-D space for 5k points.



Torus inside sphere - results for IVHD generated in 20-D space for 10k points.